

## 1.0 INTRODUCTION

### 1.1 BACKGROUND

The Defense Information Infrastructure (DII) is a defense-level enterprise effort to develop and field military systems that will meet the needs of the warfighter in a global information environment. The DII will provide “a seamless web of communications networks, computers, software, databases, applications, data, and other capabilities that meets the information processing and transport needs of DoD users in peace and in all crises, conflict, humanitarian support, and wartime roles” (DII Master Plan, page 2-1). The DII addresses systems in the command and control, intelligence, and combat support domains within the Department of Defense (DoD) and currently includes the Global Command and Control System (GCCS) and the Global Combat Support System (GCSS).

Under the DII concept, a system will be composed of software components, called segments, selected from a DII software repository and configured to meet the needs of a particular operational community. Some of the segments in the repository will offer generic functionality that can be used by multiple communities, while other segments will be targeted to particular operational requirements. To be included in the repository, segments must conform to strict standards and specifications required to support “plug and play” integration across a range of hardware platforms. When a new capability is needed in a particular mission area, segments for common functions will be assembled and customized as needed, with new software developed only to address the requirements that are unique to the mission area. Because this software is built according to the same standards and specifications as other segments, it can be added to the DII repository as another segment available for reuse.

It is critical to the overall usability of a DII system that the segments in the repository provide a user interface with a common appearance and behavior so that users can interact effectively with all of the software available in the system as well as generalize their experience to other systems in the DII. Interface standardization is particularly important as users are provided the capability to interact with a variety of complex, multi-windowed applications within a single DII system. The benefits to be gained from standardization are increased user productivity, reduced training requirements, improved system reliability, and increased efficiency in the development of individual applications as well as entire systems.

### 1.2 PURPOSE

This document defines a common appearance and behavior for the user interface of software components and information systems developed for the DII. The specifications provided here address both the “look and feel” of individual elements of the user interface as well as the design of applications and systems within the DII.

Implementing a common “look and feel” enables users to identify, remember, and predict the rules and organization of a system. By building consistency in the user interface, the user can develop an effective and efficient model of how the system works. According to Mayhew in Principles and Guidelines in Software User Interface Design (p. 97), a consistent user interface is one that provides:

- Consistent location of certain types of information on screens,
- Consistent syntax of commands in a command language,
- Similar execution of analogous operations in different applications,
- Consistent design of command names and abbreviations,
- Consistent grammatical form of error messages and instructions,
- Consistent design of captions and fields on forms and displays,
- Consistent dialog style for different functions, and
- Terminology consistent with the users’ existing vocabulary.

The specifications in the current document are intended to address these areas in sufficient detail so that when users see an object on the screen, they are able to recognize both the type of function they perform with the object and the means to perform the function.

While commonality in “look and feel” is an key element of usability, the concept of an application is central to the user understanding a system’s capabilities and how to interact with them. An application can be viewed as the software available to the user to perform a set of related tasks. This software is visible to the user as a collection of window families, each providing the functionality (in terms of objects and information) needed to perform a particular task. Because the DII provides a library of reusable segments from which to build applications, the window families with which the user interacts in a system may, in fact, be taken from multiple segments, each produced by a different developer organization. In addition, it is possible for applications to share the services provided by a segment when the applications perform common tasks. This flexibility contrasts with the traditional view of an application as a standalone entity that performs one or more self-contained tasks, all with software taken from a single source.

In a traditionally designed application, the developer’s view of the software is congruent with that of the user; i.e., the manner in which the developer designs the software matches the manner in which the user accesses and interacts with it. By contrast, when software is designed more flexibly (as occurs in the DII), the developer view may diverge from that of the user. For example, a segment may not be perceived by the user as an application at all, but rather as one of several tasks in the application from which the segment was launched. Furthermore, the user may view these tasks as providing a coherent set of mission-related capabilities when, in fact, the tasks are being performed by software taken from different sources. When divergences are possible, it is important that developer decisions in designing a software segment anticipate user expectations regarding its use when it is available as an application in a system. The specifications in this style guide are intended to provide the bridge between application and segment design and ensure that segment development accurately reflects the user view of the application.

As users interact with a system, they develop a mental model about its structure and behavior and how these relate to the operational tasks they have to perform. Developers can facilitate this process by providing a conceptual framework for the tasks performed by a segment that is consistent with the user’s view of these tasks when they are available in an application. This framework has to provide predictable cues from which users can identify and understand the structure and behavior of the overall system. These cues must be similar to those provided by traditional applications, yet accommodate the possibility that the application has been constructed from multiple segments produced by different developers. At a minimum, users should be able to map each task to the window family within which it is performed. Users should also be able to identify the window families in each application and correctly anticipate which application processes will be affected as they interact with windows that may be managed by different segments.

## **1.3 ASSUMPTIONS**

### **1.3.1 Hardware and Software Configuration**

Hardware platforms to be supported in the DII include Hewlett Packard (HP) 700 Series and Sun SPARC Series workstations, both with POSIX-compliant operating systems, as well as personal computers (PCs) running Windows NT or Windows 95. The specifications in this style guide assume that the workstation configuration for each platform includes at least one color monitor, a keyboard, and a pointing device (such as a mouse or trackball) with two or three buttons. The specifications apply to DII systems installed on HP and Sun platforms and on PCs running Windows NT; specifications for PCs running Windows 95 will be addressed in a future version of this document.

It is assumed that user interface services in DII systems are provided by X Window and Motif on UNIX-based platforms and by Microsoft Windows on PC platforms. Common Desktop Environment (CDE) is used for desktop management on UNIX-based platforms, with a user interface “look and feel” based on the CDE version of Motif. Style specifications for DII applications and systems are consistent with certification requirements set forth in documentation for the TriTeal Enterprise Desktop (TED) implementation of CDE. Because the CDE version of Motif provides significant convergence in “look and feel” with Microsoft Windows, this style guide provides an integrated set of design specifications to the extent possible, with separate specifications indicated only where the two interfaces differ.<sup>1</sup> In addition, enhancements made in CDE Motif that diverge from previous versions of Motif are indicated in footnotes.

### **1.3.2 System Users**

The primary users of DII systems are expected to be operational personnel with basic software skills but no knowledge of the operating system environment or command structure. Other user groups may include system administrators, security managers, and database administrators as required by the system. It is assumed that each system will define the functionality (i.e., specific applications) that will be available to each user category and control access to these applications during system login. For example, the functions available to a system administrator may be different than those available to normal users, and the functions available to normal users may, in turn, be configurable based on the specific tasks they perform.

### **1.3.3 Operational Environment**

It is expected that DII systems will be installed in a range of operational settings. This style guide assumes the standard environment to be an office-like workspace with normal ambient lighting and defines a default implementation for this environment. Alternative operational settings (e.g., areas with bright sunlight and significant glare, spaces where users have to remain dark adapted) will be addressed in future versions of this document.

## **1.4 SCOPE**

This style guide applies to the development of new software and the migration of existing software for those DII systems where the primary mode of interaction is through a graphical user interface (GUI); this document does not address the design of software providing a character-based interface or offer direction regarding possible migration of this software to a GUI-based interface. The specifications provided here conform with the native style defined for Motif and Windows, except where deviations are needed to accommodate operational requirements or constraints, provided that the deviations are consistent with established user interface guidelines. This document does not provide environmental or ergonomic specifications in areas such as lighting and noise or workstation design and layout.

DoD guidance concerning user interface standardization is published in the DoD Human Computer Interface Style Guide (i.e., volume 8 of the DoD Technical Architecture Framework for Information Management). The DoD document calls for the publication of style addenda when more detailed specifications at a domain or system level are needed by DoD organizations. The specifications presented here comply with guidelines in the DoD style guide and serve as the addendum for the DII. The specifications are consistent with Section 15 on user-computer interfaces in Military Standard (MIL-STD) 1472D and with Military Handbook (MIL-HDBK) 761A and incorporate guidance on user interface design published in commercial literature and available in DoD documents. The DII style guide was

---

<sup>1</sup> This style guide provides more detailed specifications for Motif-based applications than for those in Windows. Additional coverage of Windows-based design will be provided in a future version of this document.

derived from version 1.0 of the User Interface Specifications for GCCS and represents what would have been published in the next version of that document.

## **1.5 COMPLIANCE**

### **1.5.1 Assumptions Regarding Compliance**

This style guide is one of a series of related documents that define development requirements, system architecture, and engineering tools for information systems in DII. Technical implementation of DII is a cooperative effort among DoD organizations, with the Defense Information Systems Agency (DISA) defining and enforcing the DII architecture. Technical requirements for building and integrating software components are described in the DII Common Operating Environment (COE) Integration and Runtime Specification (I&RTS). This document defines eight levels of compliance representing progressively deeper integration with COE software and the runtime environment. One element of compliance addresses the extent to which a COE-based system provides a consistent “look and feel.” The definition of COE compliance in the I&RTS was taken as the starting point in specifying style-related compliance requirements for DII software. The following assumptions were made in defining these requirements:

1. Because the DII includes information systems across a range of mission areas, the style guide should provide domain-level, rather than system-level, specifications for the “look and feel” of segments in the repository, with sufficient flexibility in implementation to satisfy the range of operational requirements addressed by these systems.
2. While new software is expected to be developed in accordance with DII style specifications, most existing software is likely to predate the DII initiative and will have to evolve to COE compliance. As a result, it is important that a migration path be identified that will assist DoD organizations in transitioning to the style implementation identified as the COE target. The I&RTS defines eight compliance levels which, taken together, provide a migration path for evolving software to become fully COE compliant. The DII style guide should identify the style-specific requirements at each of these levels that will support the migration process.
3. The DII style guide should describe a “look and feel” that is within the capabilities of software developers to deliver, with successive versions of the document evolving this style in accordance with DII plans. Specific direction regarding style conformance is the responsibility of DISA and will be provided for each major release of DII software. Developers may be directed to deliver a particular conformance level based on whether the software is new or evolving. Alternatively, developers may be allowed to choose a compliance area they consider to be important or timely (e.g., to leverage style changes with other planned software upgrades), with direction only to demonstrate an increase in compliance in each new software delivery.

### **1.5.2 DII Compliance Requirements**

#### **1.5.2.1 Segment-Level Compliance**

Compliance with the specifications in the DII style guide is required in the development of all new software and the migration of existing software submitted for inclusion in the DII software repository. Software segments are expected to comply with all style guide specifications, with deviations occurring only when called for by operational requirements and approved by DISA. New software shall be developed in accordance with DII requirements and be fully compliant with all style guide specifications; existing software is expected to migrate to full COE compliance. Segments shall comply with the intent of the specifications; the style guide does not attempt to preclude all possible inappropriate, incorrect, or unacceptable implementations.

Appendix H maps style guide specifications to each of the style-related items included in the COE compliance checklist published in the I&RTS. Segments must satisfy all of the requirements for a given style-related item in order to be considered to comply with the item. As indicated in the I&RTS, the compliance level assigned to a segment is the highest numbered level where all of the checklist items have been satisfied. To be considered COE compliant in the style area, segments must demonstrate that they have satisfied all of the style-related requirements up to and including that level. Appendix I contains a User Interface Specifications (UIS) checklist that shall serve as documentation that the segment has been evaluated and complies with all relevant style requirements.

DISA will specify the style-specific requirements to be satisfied at each COE compliance level and provide direction concerning use of the UIS checklist by DoD organizations intending to submit software to the DII repository. Organizations shall deliver a completed checklist whenever they request compliance certification by DISA. If a deviation from style compliance requirements is desired, organizations shall request a waiver from DISA; if approved, waived items can be excluded from the checklist when software is assessed for compliance. DISA shall validate the accuracy of the completed checklist against the software and determine whether to certify the software at the level requested.

### **1.5.2.2 System-Level Compliance**

To be considered DII-compliant, DoD information systems shall conform to the system design specifications defined in this style guide. DoD organizations are encouraged to tailor the UIS checklist to reflect system-specific mission requirements and then to assess all of the software available in the system for compliance with the checklist. Organizations desiring to deliver a DII-compliant system but with operational requirements that dictate an alternate configuration shall obtain a waiver from DISA in order to diverge from the specifications presented here.

DoD organizations desiring to define system-specific user interface requirements not addressed in this document or previously waived by DISA shall do so by documenting them in an addendum to the DII style guide. The addendum shall provide detailed guidance concerning user interface features not specifically addressed within the style guide as they relate to the needs of the user community for whom the system is intended. The addendum shall maintain consistency with the user interface style defined by DII, extending the scope and content of the specifications as needed to address unique user requirements. The addendum shall be written so that it supplements, rather than duplicates, information already included in the DII style guide. Republication of the entire DII style guide, with changes to reflect system-unique requirements, is strongly discouraged.

### **1.5.2.3 Minimum Style Requirements for COE Compliance**

While DII style specifications are intended to provide sufficient flexibility in implementation to satisfy a range of operational requirements, there are some basic elements of a Motif- or Windows-based user interface that shall not be modified in DII software. These elements are:

- The hotspot of the pointer indicates the locus of user input with the pointing device.
- The location cursor indicates the locus of user input from the keyboard.
- Only one window has input focus at any time.
- Window management operations affect only the windows in a window family.

The above elements are central to the fundamental paradigm underlying a graphical interface and are considered essential to the DII integration process at the user interface level. DII software must satisfy these minimum style requirements in order to be considered COE compliant at level 1 (see appendix H). Requests for waivers to diverge from these requirements shall be directed to DISA and considered on a case-by-case basis.

### 1.5.3 Modifications to the Style Guide

The DII style guide shall be modified as needed to ensure continued compliance with Motif, Windows, and CDE style direction, maintain consistency with DoD policy and publications on user interface design, and address new user interface technologies. Revisions to the style guide shall coincide with major DII software releases and document the continued evolution of the DII user environment. Requests to modify DII style specifications shall be submitted for consideration in accordance with configuration management procedures established for the document by DISA. Requests to modify the implementation of a user interface feature in DII software shall be directed to the organization with configuration management responsibility for the software.

## 1.6 DOCUMENT OVERVIEW

The remainder of this document describes the basic elements of the user interface, explains system and application design in the DII, and addresses some additional topics in user interface design. Unless otherwise indicated, the specifications apply to both Motif and Windows applications. If an implementation is available in only one of the GUIs, it is identified as such and is either indented within a paragraph or included in parentheses within a sentence.

Sections 2 and 3 describe the input devices available to users and the manner in which these devices are used to interact with the applications in a system.

Sections 4, 5, and 6 provide detailed specifications describing the appearance and behavior of windows, menus and controls in an application.

Section 7 is an overview of system design with CDE and describes two models for application design in the DII.

Section 8 provides design guidelines for primary, secondary, and dialog windows in an application, and section 9 presents generic guidelines concerning the presentation of information in these windows.

Sections 10 and 11 address the design of task-specific windows and availability of user support resources.

Sections 12, 13, and 14 provide guidance related to user interface internationalization, design of on-line user documentation, and user interface functionality in common support applications. The guidelines in these sections are to be implemented in the application but are not considered in determining style compliance and are not included in the checklist.

Appendix A and B identify the keyboard functions assigned to virtual keys in Motif and Windows and map these keys to the keyboards for the standard hardware platforms in the DII.

Appendix C defines standard vocabulary, mnemonics, and accelerators for common actions in DII applications.

Appendix D provides developer notes that translate style specifications into resource settings and configuration files for CDE, Motif, and Windows; appendix E lists the fonts for which font aliases are available in CDE.

Appendix F lists acronyms and abbreviations used in the document, while Appendix G maps the terminology used in the current style guide to that in CDE Motif and Windows documentation.

Appendix H identifies the style requirements for COE compliance; appendix I contains a checklist for assessing the extent to which software satisfies these requirements.

## 1.7 TYPOGRAPHIC CONVENTIONS

The following typographic conventions are used in the style guide:

- a. Push button names (e.g., Save button) and actions (e.g., Cancel, OK) are capitalized as are menu titles (e.g., File) and options (e.g., Minimize, Maximize). In addition, the names of specific windows (e.g., Map Countries window) and applications (e.g., Fuel Calc application) are capitalized.
- b. The names of buttons on the pointing device are capitalized (e.g., the Select button); when the name refers to a virtual button, it is preceded by "B" (e.g., BSelect).
- c. The names of keys on a keyboard are capitalized (e.g., Return); when the name refers to a virtual key, it is shown in brackets (e.g., <Return>).
- d. Simultaneous key combinations are indicated by presenting the key names separated by a plus; for example, Ctrl+Prev means that users hold down the Ctrl key and then press the Prev key. The same meaning applies when virtual key names are presented together (e.g., <Shift><Esc>).

## 1.8 REFERENCES

The specifications in this document are based on the guidelines, standards, and style guides listed below.

### Government Documents

Air Force Intelligence Agency. Air Force Intelligence Data Handling System (IDHS) Style Guide. Air Force Intelligence Agency, Washington, D.C., 1990.

Avery, L.W. & Bowser, S.E. (eds.) Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface, Version 2.0. Pacific Northwest Laboratory for the U.S. Army Tactical Command and Control System Experimentation Site, Fort Lewis, WA, 1992.

Bowen, C.D. Theater Battle Management (TBM) Human Computer Interface (HCI) Specification, Version 1.2. The MITRE Corporation, Bedford, MA, 1995.

Defense Information Systems Agency. Defense Information Infrastructure (DII) Integration and Runtime Environment Specification (I/RTES), Version 2.0 (Preliminary), 1995.

Defense Information Systems Agency. Defense Information Infrastructure Master Plan, Version 3.0, 1995.

Defense Information Systems Agency. Department of Defense Technical Architecture Framework for Information Management. Volume 2: Technical Reference Model and Standards Profile Summary, Version 2.0, 1994.

Defense Information Systems Agency. Department of Defense Technical Architecture Framework for Information Management. Volume 8: Department of Defense Human Computer Interface Style Guide, Version 3.1, 1994.

Defense Information Systems Agency. User Interface Specifications for the Global Command and Control System (GCCS), Version 1.0, 1994.

Department of the Air Force, Standard Systems Center (SSC). Graphical User Interface (GUI) Standards, Volume 1, 1993.

Department of Defense. Military Handbook 761A. Human Engineering Guidelines for Management Information System. Department of Defense, Washington, D.C., September 1989.

Department of Defense. Military Standard 1472D. Human Engineering Design Criteria for Military Systems, Equipment, and Facilities. U.S. Army Missile Command, Huntsville, AL, 1989.

Department of Defense Intelligence Information Systems Management Board. Department of Defense Intelligence Information System (DODIIS) Profile of the DoD Technical Reference Model for Information Management. Bolling Air Force Base, Washington, D.C., 1993.

Naval Integrated System for Exploitation (NISE) Workstation Requirements Specification, 1991.

Operations Directorate Graphical User Interface Standards. Version 1.0. Prepared by the Joint DO/DT GUI Standards Working Group, 1994.

Smith, S.L. & Mosier, J.N. Guidelines for Designing User Interface Software (ESD0TR086-278). USAF Electronic Systems Center, Hanscom AFB, MA, 1986.

#### Non-Government Documents

Apple Computer, Inc. Guide to Macintosh Software Localization. Reading, MA: Addison-Wesley Publishing Co., 1992.

Apple Computer, Inc. Human Interface Guidelines: The Apple Desktop Interface. Reading, MA: Addison-Wesley Publishing Co., 1987.

CDE Documentation Group. Common Desktop Environment 1.0 User's Guide. Reading, MA: Addison-Wesley Publishing Co., 1995.

Ferguson, P.M. Motif Reference Manual for OSF/Motif Release 1.2. Sebastopol, CA: O'Reilly & Associates, Inc., 1993.

Fowler, S.L. & Stanwick, V.R. The GUI Style Guide. Boston, MA: Academic Press, Inc., 1995.

Galitz, W.O. User-Interface Screen Design. Boston, MA: QED Publishing Group, 1993.

Gardiner, M.M. & Christie, B. (eds.) Applying Cognitive Psychology to User-Interface Design. Chichester: John Wiley & Sons, 1987.

IEEE Recommended Practice for Graphical User Interface Drivability (Unapproved Draft 2), March 1992.

Kobara, S. Visual Design with OSF/Motif. Reading, MA: Addison-Wesley Publishing Co., 1991.



Mayhew, D.J. Principles and Guidelines in Software User Interface Design. Englewood Cliffs, NJ: PTR Prentice Hall, 1992.

Madell, T., Parsons, C. & Abegg, J. Developing and Localizing International Software. Englewood Cliffs, NJ: Prentice Hall, 1994.

Marcus, A., Smilonich, N. & Thompson, L. The Cross-GUI Handbook for Multiplatform User Interface Design. Reading, MA: Addison-Wesley Publishing Co., 1995.

Microsoft Corporation. The Windows Interface: An Application Design Guide. Redmond, WA: Microsoft Press, 1992.

O'Donnell, S.M. Programming for the World: A Guide to Internationalization. Englewood Cliffs, NJ: Prentice Hall, 1994.

Open Software Foundation. OSF/Motif Style Guide. Release 1.2. Englewood Cliffs, NJ: Prentice Hall, 1992.

Open Software Foundation. Motif Style Guide, Release 2.0 (Beta draft), 20 January 1994.

Open Software Foundation. OSF/Motif User's Guide. Release 1.2. Englewood Cliffs, NJ: Prentice Hall, 1992.

Root, R.W. & McFarland, A.D. Graphical User Interface Design Guidelines for Bellcore Software Products. Issue 1. Bellcore/Bell Communications Research, Piscataway, NJ, 1993.

TriTeal Corporation. TriTeal Enterprise Desktop (TED) 4.0 Advanced User's and System Administrator's Guide. Carlsbad, CA: TriTeal Corporation, 1995.

TriTeal Corporation. TriTeal Enterprise Desktop (TED) 4.0 Enhancements. Carlsbad, CA: TriTeal Corporation, 1995.

TriTeal Corporation. TriTeal Enterprise Desktop (TED) 4.0 Programmer's Guide. Carlsbad, CA: TriTeal Corporation, 1995.

TriTeal Corporation. TriTeal Enterprise Desktop (TED) 4.0 Programmer's Overview. Carlsbad, CA: TriTeal Corporation, 1995.

TriTeal Corporation. TriTeal Enterprise Desktop (TED) 4.0 Style Guide and Certification Checklist. Carlsbad, CA: TriTeal Corporation, 1995.

Uren, E., Howard, R. & Perinotti, T. Software Internationalization and Localization: An Introduction. New York: Van Nostrand Reinhold, 1993.

User Interface Design With OSF/Motif. Open Software Foundation Training Course, Version 1.2, 1992.

The specifications presented here make reference to the following additional documents:

Defense Intelligence Agency. Standard Military Graphics Symbols Manual (DIAM 65-XX) (Draft). Defense Intelligence Agency, 1990.

Department of the Army. Army Field Manual FM 101-5-1. Operational Terms and Symbols. U.S. Army Combined Arms Center, Fort Leavenworth, KS, 1985.

Department of the Army. AR 310-50. Authorized Abbreviations and Brevity Codes. Army UPDATE Publications, Washington, D.C., 1985.

Department of Defense. Military Standard 12D. Abbreviations for Use on Drawings, Specification Standards, and in Technical Documents, 1981.

Department of Defense. Military Standard 411E. Air Crew Station Alerting Systems, 1991.

Department of Defense. Military Standard 783D. Legends for Use in Air Crew Stations and on Airborne Equipment, 1984.

Department of Defense. Military Standard 2525. Common Warfighting Symbolology, Version 1, 1994.

North Atlantic Treaty Organization Standardization Agreement 2019. Military Symbols for Land-Based Systems, 1990.

NATO Standardization Agreement 4420. Display Symbolology and Colours for NATO Maritime Units, 1990.

Standards relating to the design of workstations, associated furniture, and the facilities in which they are placed can be found in MIL-STD-1472D and the following document:

American National Standards Institute. National Standard for Human Factors Engineering of Visual Display Terminal Workstations. Santa Monica, CA: The Human Factors Society, Inc., 1988.

DoD guidance on these topics can be found in the following document:

Department of Defense. DoD Human Computer Interaction (HCI) Concept Plan (Revised), 1995.